Future Computing and Informatics Journal

Volume 2
Issue 1 (2020) *2017, Volume 2, Issue 1*

Article 5

2017

Forecasting of nonlinear time series using ANN

Ahmed Tealab
Computer Science Department, Institute of Statistical Studies and Research, Cairo University,
a.tech.gouda@gmail.com

Hesham Hefny
Computer Science Department, Institute of Statistical Studies and Research, Cairo University, Giza,
hehefny@ieee.org

Amr Badr
Computer Science Department, Faculty of Computers and Information, Cairo University, Giza, Egypt,
a.badr.fci@gmail.com

Available online at www.sciencedirect.com

ScienceDirect

Future Computing and Informatics Journal 2 (2017) 39–47
http://www.journals.elsevier.com/future-computing-and-informatics-journal/

Forecasting of nonlinear time series using ANN

Ahmed Tealab [a,*], Hesham Hefny [a], Amr Badr [b]

[a] Computer Science Department, Institute of Statistical Studies and Research, Cairo University, Giza, Egypt
[b] Computer Science Department, Faculty of Computers and Information, Cairo University, Giza, Egypt

Received 19 March 2017; accepted 14 May 2017
Available online 3 July 2017

Abstract

When forecasting time series, it is important to classify them according linearity behavior that the linear time series remains at the forefront of academic and applied research, it has often been found that simple linear time series models usually leave certain aspects of economic and financial data unexplained. The dynamic behavior of most of the time series in our real life with its autoregressive and inherited moving average terms issue the challenge to forecast nonlinear times series that contain inherited moving average terms using computational intelligence methodologies such as neural networks. It is rare to find studies that concentrate on forecasting nonlinear times series that contain moving average terms. In this study, we demonstrate that the common neural networks are not efficient for recognizing the behavior of nonlinear or dynamic time series which has moving average terms and hence low forecasting capability. This leads to the importance of formulating new models of neural networks such as Deep Learning neural networks with or without hybrid methodologies such as Fuzzy Logic.

Keywords: Forecasting; Nonlinear time series; Neural networks; Moving averages

1. Introduction

Although the forecasting of time series has generally been made under the assumption of linearity, which has promoted the study and use of linear models such as the autoregressive (AR), Moving Averages (MA), autoregressive moving averages (ARMA) and autoregressive integrated moving averages (ARIMA) [1.2], it has been found that in reality the systems often have unknown nonlinear structure [3]. To address this problem type, several nonlinear models have been proposed, such as the bilinear models, autoregressive conditional heteroskedasticity (ARCH) and its extensions, smooth transition autoregressive (STAR), nonlinear autoregressive (NAR), wavelet networks and artificial neural networks (ANN) [1–7].

With regard to the ANN, it is that its theory is very wide, and it has been applied in modeling and forecasting data from different knowledge areas [1–3,8–14]; however, in the literature there is a large part of the proposed ANN models that are exclusively based on a nonlinear autoregressive structure, and only a few of them considered the generating process of the nonlinear time series that has in addition to the autoregressive, a moving averages component. To address this case, some authors suggest using the neural network NARMA and the autoregressive neural network ARNN of high order; in Refs. [15,16] present such specific cases.

However, in reviewing the relevant literature finds that:

- The theory of NARMA (p,q) model considers that the process of data generation corresponds to a nonlinear structure with both autoregressive and moving average components; which is done by ignoring the autoregressive component (making $p = 0$) to obtain a nonlinear model of moving averages (NLMA); however, in the literature there

* Corresponding author.
E-mail addresses: a.tech.gouda@gmail.com (A. Tealab), hehefny@ieee.org (H. Hefny), a.badr.fci@gmail.com (A. Badr).
Peer review under responsibility of Faculty of Computers and Information Technology, Future University in Egypt.

http://dx.doi.org/10.1016/j.fcij.2017.05.001

are no studies that examine the capability of forecasting of NARMA (0,q) when it is applied in a nonlinear time series that present an inherent MA component.
- There is no evidence, reported that a nonlinear MA model can be approximated by a nonlinear infinite order AR model, like what happens in the case of linear models when they meet certain invertibility conditions.

The objective of this research is to answer the research questions presented below in order to clarify the above gaps:

1. Can a nonlinear high order AR model, represented by ARNN network, be well approximated to nonlinear reduced order MA model?
2. When in a NARMA that assumes there is no autoregressive process, can be predicted adequately a nonlinear time series containing inherent moving averages components?

These questions will be resolved on the basis of the approach of the invertibility of the nonlinear MA models and the use of experimental data simulations.

The importance and originality of this work is based on the fact that to date there is no evidence in the reviewed literature of studies that analyze and identify the problems that arises when modeling and forecasting time series with inherent MA components using neural networks. The article is organized as follows: in sections 2 and 3 present the nonlinear MA model, and the NARMA and NAR neural networks, respectively. Subsequently, in section 4, it shows the methodology used and the results obtained to assess the capacity of these networks to predict nonlinear time series with MA component. Section 5 presents the obtained results, while in section 6 provides answers to the research questions raised. Finally, it is the conclusion in section 7.

2. Nonlinear moving average model

In the nonlinear moving average model of order q, denoted as NLMA (q), the current value of the time series, y_t, is a nonlinear function known as $h(.)$ of the q past innovations $\{\varepsilon_{t-1}, \cdots, \varepsilon_{t-q}\}$ and the current innovation ε_t.

This is:

$$y_t = \varepsilon_t + h\left(\varepsilon_{t-1}, \cdots\cdots, \varepsilon_{t-q}; \theta\right), \quad t = 1, 2, 3, \tag{1}$$

where θ represents the parameters vector of function $h(.)$ and y $\{\varepsilon_t\}$ is a sequence of independent random variables which are identically distributed, centered at zero and with constant variance.

Depending on the form of the function h $(\cdot)$, the following NLMA models have been proposed:

- Polynomial moving averages proposed by Robinson [17].
- Asymmetric moving averages proposed by Brännäs and Ohlsson [18].
- Nonlinear response moving averages with long scope proposed by Robinson and Zaffaroni [19].
- Nonlinear integrated moving average by Engle and Smith [20].

Different to the nonlinear autoregressive model (NAR), the NLMA model has been little explored, both empirically and theoretically. This is due, in part to the difficulty to establish the invertibility property model [21]; that property refers to the possibility of rebuilding innovations ε_t from the observations y_t, assuming that the true model is known. However, Chen and Wang [22] reached that the NLMA model can become locally invertible; that can be done by set the initial conditions that allow the innovations reconstruction asymptotically from the observations.

The fact that the NLMA model is not globally invertible makes it, at least theoretically, not equivalent to a high order NAR model, as if it happens in the case of linear one. It is important to verify the invertibility of NLMA model to ensure that it is appropriate for the forecasting purposes and also make its diagnosis possible.

3. Neural networks models associated with moving averages components

Mathematically, a neuron is a nonlinear function, bounded and parameterized in the form [23]:

$$o = f\left(x_1, x_2, \ldots, x_n; \omega_1, \omega_2, \ldots., \omega_p\right) = f(\boldsymbol{x}; \boldsymbol{\omega})$$

where:

- $x = (x_1, x_2, \ldots, x_n)$ is the entry vector of variables into the neuron.
- $\omega = (\omega_1, \omega_2, \ldots, \omega_p)$ is the weight (parameters) vector associated with the inputs of the neuron.
- $f(\cdot)$ is a nonlinear activation function.

In turn, an artificial neural network is defined as a composition of nonlinear functions of the form:

$$y = g_1 \circ g_2 \circ \ldots \circ g_N\left(f_1(\boldsymbol{x}; \boldsymbol{\omega}), f_2(\boldsymbol{x}; \boldsymbol{\omega}) \ldots, f_p(\boldsymbol{x}; \boldsymbol{\omega})\right)$$

where:

- y is the response variable or output of the artificial neural network.
- g_1 for $i = 1,\ldots.,N$ are nonlinear functions.
- $f_j(x; \omega)$ for $j = 1,\ldots,p$ are functions defined as in (1).
- N represents the number of hidden layers in the network.
- p denotes the number of neurons in the hidden layers.
- The symbol $\circ$ between functions indicates the operation composition.

The neural networks, according to its architecture and interconnection between neurons, can be classified into two classes: feed-forward networks and feed-back (recurrent) networks. The feed-forward network, also known as static, constitutes a nonlinear function of their entries and is

represented as a set of interconnected neurons, in which information flows only in the forward direction, from inputs to outputs. Specifically, in Ref. [24] a feed-forward network model, with a single output neuron and q hidden layers, is defined as follows:

$$o_t = \Phi\left(\beta_0 + \sum_{i=1}^{q} \beta_i \Psi\left(\alpha_i + \sum_{j=1}^{n} \omega_{ij} x_{j,t}\right)\right) =: f(x_t;\theta) \tag{2}$$

where

- O_t is the estimator of the target variable y_t.
- $x_t = (x_{1,t},\ldots,x_{n,t})$ are input variables in time measures t.
- $\Phi(\cdot)$ and $\Psi(\cdot)$ are the activation functions of the neural network.

$\Theta = (\beta_0, \beta_1,\ldots \beta_q, \alpha_1,\ldots, \alpha_q, \omega_{11},\ldots, \omega_{qn})$ represents the parameters vector of the neural network, which is calculated based on the minimization of the sum of squared differences

$$\sum_{t=1}^{n} (y_t - \hat{o}_t)^2$$

It is noteworthy that the kind of neural networks is more studied and applied in the literature, mainly due to they are a universal function approximator [25–27]; and moreover, in practice they are more simple networks in their implementation and simulation. Meanwhile, the feed-back network, also known as dynamic or recurrent, its architecture is characterized by cycles: the outputs of the neurons in a layer can be inputs to the same neuron or inputs to neurons of previous layers. For more information of this type of network is suggested to check [23] and [28]. Below are described special cases of these types of networks: the autoregressive neural network ARNN, which is of type feed-forward and recurrent neural network NARMA.

3.1. Autoregressive neural network (ARNN)

The nonlinear autoregressive model of order p, NAR (p), defined as:

$$y_t = h(y_{t-1},\ldots,y_{t-p}) + \varepsilon_t \tag{3}$$

is a direct generalization of linear AR model, where $h(\cdot)$ is a nonlinear known function. It is assumed that $\{\varepsilon_t\}$ is a sequence of random independent variables and identically distributed with zero mean and finite variance σ^2.

The autoregressive neural network (ARNN), is a feed-forward network constitutes a nonlinear approximation $h(\cdot)$, which is defined as:

$$\begin{aligned} \hat{y}_t &= \hat{h}(y_{t-1},\ldots,y_{t-p}) \\ \hat{y}_t &= \beta_0 + \sum_{i=1}^{I} \beta_j f\left(\alpha_i + \sum_{j=1}^{p} \omega_{ij} y_{t-j}\right) \end{aligned} \tag{4}$$

where $f(\cdot)$ function is the activation function and $\Theta = (\beta_0, \beta_1,\ldots \beta_q, \alpha_1,\ldots, \alpha_q, \omega_{11},\ldots, \omega_{qn})$ is the parameters vector.

3.2. Recurrent neural network NARMA

A generalized linear model ARMA in the nonlinear case is given by

$$y_t = h(y_{t-1},\ldots,y_{t-p},\varepsilon_{t-1},\ldots,\varepsilon_{t-q}) + \varepsilon_t$$

where $h(\cdot)$ is a known nonlinear function and $\{\varepsilon_t\}$ is defined as in (3). This model is called NARMA (p,q). Since the sequence $\varepsilon_{t-1},\ldots, \varepsilon_{t-q}$ is not directly observable, then you must find one $\hat{y}_t$ using recursive estimation algorithm that considers the following calculations:

$$\hat{y}_t = h(y_{t-1},\ldots,y_{t-p},\hat{\varepsilon}_{t-1},\ldots,\hat{\varepsilon}_{t-q}) \tag{5}$$

$$\hat{\varepsilon}_j = y_{t-1} - \hat{y}_j, \quad j = t-1,\ldots,t-q \tag{6}$$

under appropriate initial conditions [16]. By considering the approximation in (5) and (6) the recurrent neural network model NARMA (p,q) can be expressed using the recurrent network:

$$\hat{y}_t = a_0 + \sum_{j=1}^{h} a_j g\left(\beta_{0j} + \sum_{i=1}^{p} \beta_{ij} y_{t-i} + \sum_{i=p+1}^{p+q} \beta_{ij} \hat{\varepsilon}_{t+p-i}\right) \tag{7}$$

where $\hat{\varepsilon}_{t+p-i} = y_{t+p-i} - \hat{Y}_{t+p-i}$.

By observing the mathematical formulation of the model (7), it can be considered as an alternative to a nonlinear time series model with an inherent moving averages component is to use a NARMA $(0,q)$ model. This observation will be discussed in the following section.

4. Used methodology

The evaluation of the ability to forecast NARMA (p,q) neural networks models and ARNN (p) was performed using two sets of experimental data from the models described in Table 1. In Model 1 $\{\varepsilon_t\}$ is defined as in (3), and corresponds to the NLMA (2) model reviewed by Zhang et al. [29]. On the other hand the Model 2 was considered by Burges and Refenes [15] to illustrate the use of neural networks with feed-back error under the expectation–maximization, EM variant algorithm in the training process.

Note that the two models do not contain autoregressive terms (do not consider past y_t values), also correspond to different levels of complexity of the function $h(\cdot)$ defined in (1). 100 time series were generated from each model. Of which, in each series generated, the first observations were

Table 1
Data generation models.

Model	Model structure
1	$y_t = \varepsilon_t - 0.3\varepsilon_{t-1} + 0.2\varepsilon_{t-2} + 0.4\varepsilon_{t-1}\ \varepsilon_{t-2}$
2	$y_t = \varepsilon_t + 0.5\varepsilon_{t-1} + 06\varepsilon_{t-1}\ \varepsilon_{t-2}$

used to estimate the model parameters and the remaining were used as validation set. In Fig. 1 of the series Model 1 is plotted with $n = 360$ observations. In the data generation process, different beginnings with random sampled distribution N(0; 1,5) are used with the error term of model 1, and it is assumed in the Model 2 that $\varepsilon_{-1} = \varepsilon_{-2} = 0$ and $y_0 = \varepsilon_0 = \text{rand}()$, where rand() return a standard uniform random number.

The experiments focused on two aspects: (i) Analysis the ability to capture all the nonlinear moving averages process using a recurrent neural network NARMA (0,q) or ARNN(p) with large enough p, (for which the Model 1 was used), and (ii) compare results obtained with any of the networks considered in this work with those found in the literature to NLMA model processes. In this case the Model 2 was used, and compared the Burges and Sayings results [15] obtained by a ARNN(p) network. In that case, the methodology used for each model has some distinctive aspects:

Model 1:

- Different sample sizes are considered for $n = \{100; 200; 360\}$ and data rates for network training (50, 65 and 80), to examine the effect of their election on the predicted values.
- For the ARNN model values were examined with large lags of $p = \{10; 15; 25; 50; 100\}$ for the purpose to answer the first research question.
- The network structure was considered to be used based on the results found by Zhang et al. [29], who via simulation show that the best network structure corresponds to a hidden layer with a maximum of two neurons. The objective function was minimizing the mean square error (MSE).
- In the case of NARMA model, in addition to the structure of previous network, the following settings for the moving averages process were considered that $p = \{1; 2; 3; 4; 5; 6; 7; 8; 9; 10\}$.
- A set of 150 additional observations was generated and used as test data.

Model 2: It is considered the same experimental conditions employed by Burges and Refenes [15] in order to be able to compare the results:

- The size of the series was 400 observations, of which the initial 70% is used to train the network and the remaining 30% for validation.

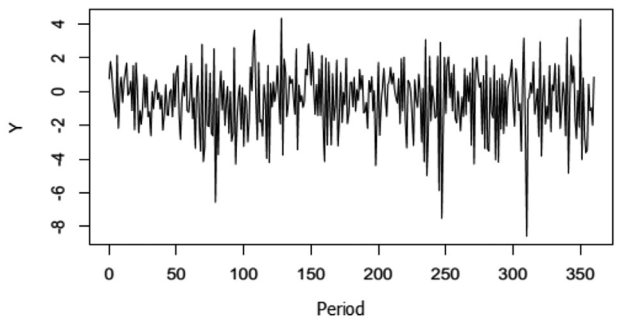

Fig. 1. Example of time series generated by Model 1.

- The objective function was to minimize the normalized means square error (NMSE).
- All networks used were of one hidden layer with four neurons.
- The following lags values $p = \{10; 25; 50\}$ were considered.
- 100 additional data were generated, and were taken as test data.

In both models the activation function used was the logistics, for each training, initial weights and biases of the network were generated from a continuous uniform distribution in the range $(-5; 5)$. Also, the choice of the best model was performed by taking into account the 100 series and different configurations of the network, under the cross-validation procedure suggested by Zemouri et al. [30], namely:

1. Made from i = 1 to M = 1000 times from different starting points:
   - Train the network using the training data.
   - Validate the trained network using the *n.val* validation data. Calculate the forecasting mean error $E(i)$ and standard deviation $std(i)$ on the validation set:

$$E(i) = \frac{1}{n.val}\sum_{j=1}^{n.val}\left(y_j - \hat{y}_{j^-}\right) \tag{9}$$

$$std(i) = \frac{1}{n.val}\sum_{j=1}^{n.val}\left(y_j - \hat{y}_{j-}\right)^2 \tag{10}$$

2. Calculate the following measures to evaluate the forecasting performance of the network:
   - $M1 = \overline{E} = 1/M\sum_{i=1}^{M} E(i)$. It corresponds to an estimate of the average of the overall forecasting mean errors, and evaluates the proximity between the predicted and actual values. If $M1 = 0$, then probability that the forecasting is centered on the actual data is very high.
   - $M2 = \overline{std} = 1/M\sum_{i=1}^{M} std(i)$. It is used for measuring forecasts accuracy (in terms of variability). The ideal value is $M2 = 0$, because it indicates that there is a significant probability that the predicted values are not scattered (i.e.; they have low variability).
   - $M3 = \sqrt{1/M\sum_{i=1}^{M}[E(i) - \overline{E}]^2} + \sqrt{1/M\sum_{i=1}^{M}[std(i) - \overline{std}]^2}/2$. It is used to indicate whether the training process of the network is repeatable (in which case $M = 3$), so that you always get the same structure of the neural network in each run of the training process, regardless of the initial values.
   - $M4 = 1/M1 + M2 + M3$. It is to examine the accuracy of the forecast. If the outputs of the network are very close to the actual values, then the measures $M1$, $M2$ and $M3$ are close to zero, and in that case $M4$ will

take very large values, so that $M4 >> 0$ is the ideal value to have forecasts confidence.

3. Perform the verification using the test data: Select the best candidate network as it having the higher $M4$ value and lower $M1$, $M2$, and $M3$ values on the validation set. This will avoid over-fitting and under-fitting problems. Finally, the M is made for that network is reached and the model with the lowest $E(i)$ is selected.
4. Perform data verification test: calculate $E(i)$ and $std(i)$ for each selected configuration (one for each series in question). Choose the model that provides lower $E(i)$.

The above measures were used to validate the accuracy of the results obtained from the network under study.

5. Results

The obtained results are presented below for each considered model.

5.1. Model 1

Figs. 2–4 show the values obtained for the measures $M1-M4$ on the validation set of ARNN network for each sample size under different numbers and considered training lags percentages. In turn Fig. 5 contains the values of performance measures $E(i)$ and $std(i)$ obtained in the validation set. Table 2 shows the results found for the ARNN network,

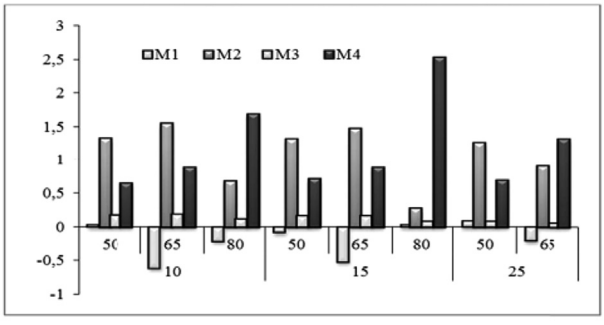

Fig. 2. Performance measures for the ARNN model with $n = 100$, $p = \{10, 15, 25\}$ and training percentage (50, 65, 80).

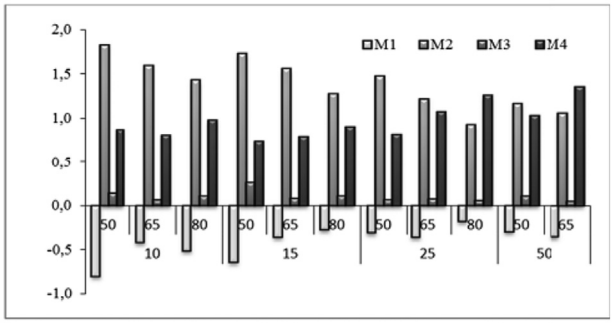

Fig. 3. Performance measures for the ARNN model with $n = 200$, $p = \{10, 15, 25, 50\}$ and training percentage (50, 65, 80).

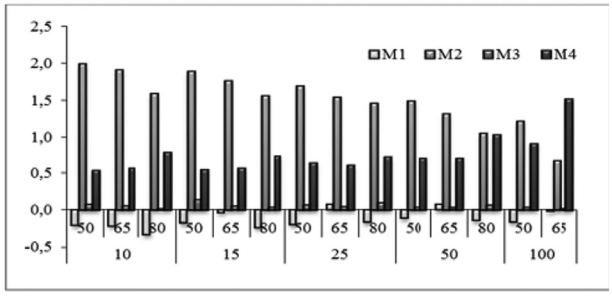

Fig. 4. Performance measures for the ARNN model with $n = 360$, $p = \{10, 15, 25, 50, 100\}$ and training percentage (50, 65, 80).

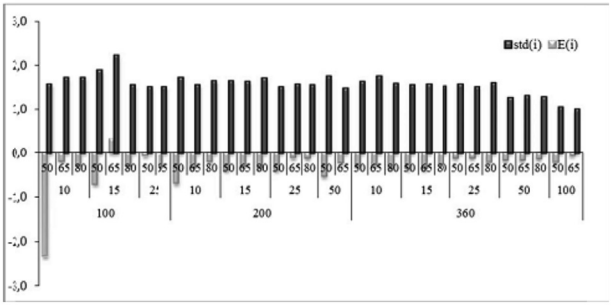

Fig. 5. $E(i)$ and $std(i)$ measures in the validation set for ARNN model, according to the sample size, lags and training percentage.

the test data under nine considered scenarios and the values of large lags $p = \{10; \text{fifteen}; 25; 50; 100\}$. The first column contains the sample size, the second number of lags p, and the last three columns show the measures $E(i)$ and $std(i)$ values

Table 2
Performance measures for the ARNN model with the test data.

n	p	Measure	Percentage of training		
			50	65	80
100	10	$E(i)$	−2.334	−0.1958	−0.2934
		$std(i)$	1.5836	1.7273	1.7324
	15	$E(i)$	−0.7113	0.3311	−0.3016
		$std(i)$	1.8923	2.2368	1.5623
	25	$E(i)$	−0.3702	−0.316	*
		$std(i)$	1.5233	1.5139	*
200	10	$E(i)$	−0.6903	−0.339	−0.2041
		$std(i)$	1.7291	1.5689	1.6601
	15	$E(i)$	−0.3939	−0.3065	−0.3379
		$std(i)$	1.6468	1.6451	1.7154
	25	$E(i)$	−0.3945	−0.1167	−0.129
		$std(i)$	1.5177	1.5716	1.5575
	50	$E(i)$	−0.5299	−0.2362	*
		$std(i)$	1.756	1.4851	*
360	10	$E(i)$	−0.3284	−0.299	−0.346
		$std(i)$	1.6458	1.7647	1.5909
	15	$E(i)$	−0.3678	−0.371	−0.2601
		$std(i)$	1.559	1.5777	1.5391
	25	$E(i)$	−0.1201	−0.123	−0.2232
		$std(i)$	1.5785	1.5136	1.6092
	50	$E(i)$	−0.1744	−0.1713	−0.1388
		$std(i)$	1.2746	1.3208	−1.2823
	100	$E(i)$	−0.2222	**−0.05965**	*
		$std(i)$	1.06824	**1.01172**	*

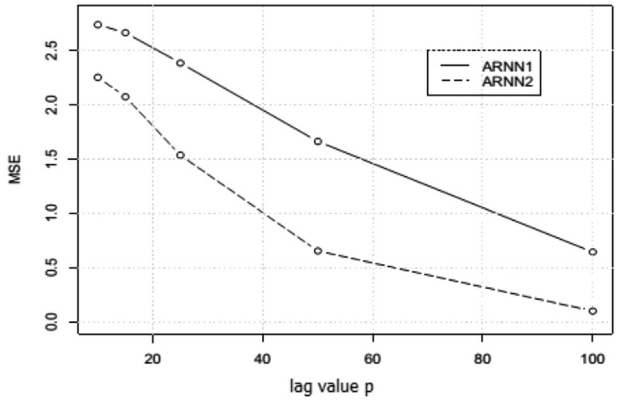

Fig. 6. Number of lags of the nonlinear model versus the MSE of the ARNN network with one (ARNN1) and two (ARNN2) nodes in the hidden layer.

found for the test percentage of training sets. In this table, the * symbol indicates that the value of the lag p is greater than the size of the sample for all validation set, so you cannot examine the ability of forecasting in this group of data.

From Figs. 2–5, it appears that whatever the value of the gap, there is a direct relationship between the percentage of training and forecast accuracy. Regarding the reproducibility of the model, it is observed that networks generally adjusted always satisfy this condition. Finally, the greater forecast accuracy is obtained by combining the maximum lag allowed to the maximum percentage of training and sample size. Note to month, that the quality of the forecast, in terms of declining values $E(i)$ and $std(i)$ is better as $p \rightarrow \infty$. That in turn makes the overall mean and forecast accuracy converges to their ideal values.

In addition, Table 2 and Figs. 2–4 follow that the number of lags selected in the final ARNN model depends on the size of the series and the percentage of data used for network training: for the network to be able to predict adequately, it is necessary to choose the maximum number of lags allowed and the largest set of training; which leads to expect that the use of ARNN networks to forecast series with inherent MA component, tends to suffer from parameterization problems. This is confirmed by examining the behavior of the MSE according to the number of lags and layers of the network. It was observed that in the way of increasing the order of the nonlinear AR model, the MSE tends to decrease regardless of the nodes considered; however, minors MSE is obtained when considering the network with two nodes in the hidden layer (see Fig. 6).

The best result found for the ARNN network (in terms of better measures results on the test data) was obtained when considering 360 observations, of which 65% were used to train the network with the maximum number of lags (100) and 2 nodes in the hidden layer. However, it is not able to capture all the nonlinear process of moving averages (see graphic (a) in Fig. 7).

Moreover, the results found on the predictive ability of the recurrent neural network NARMA with presence of moving averages are shown in Table 3 and the graph (b) of Fig. 7.

In Table 3, the first column shows the sample size, and the last three columns shown for each percentage of the following training results: selected configuration (number of lags p and number of nodes in the hidden layer◊) measurements values obtained for $M1-M4$ on the validation set, and the $E(i)$ and $std(i)$ values for the whole test and the last three columns show the values obtained from these measurements for each training percentage.

In this table it is concluded that the NARMA network requires considering large sample sizes to fit models that capable of reduce the forecasts heterogeneity in test set. Likewise, as in the networks ARNN, the percentage of data used for training the network has a direct relationship with the accuracy of the forecast, for any sample size. It was found that the best outcome for the NARMA network (in terms of the measures on the test data) was provided considering two nodes in the hidden layer, $q = 2$ lags and 360 observations, of which 80

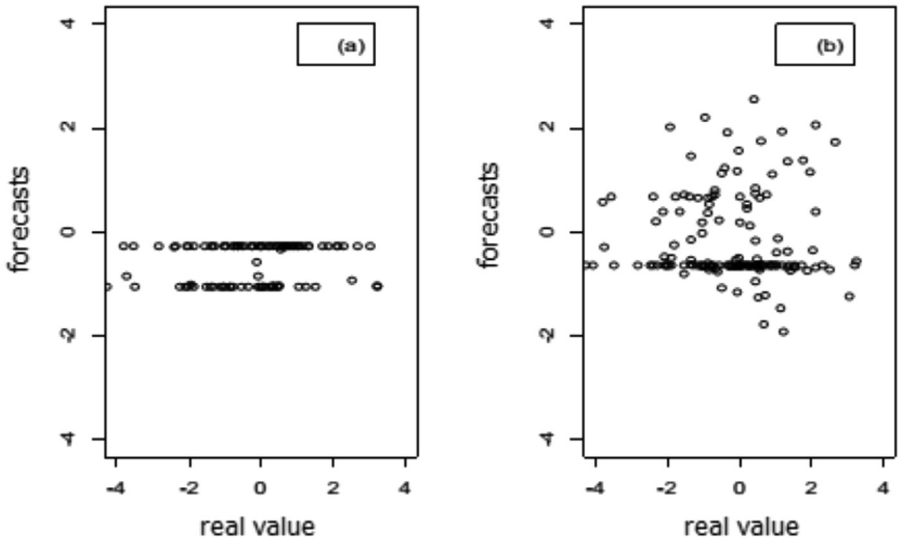

Fig. 7. Comparison between the test data and their found forecasts with the best network (a) ARNN (100) and (b) NARMA ($q = 2$, $k = 2$).

Table 3
Measures of performance for the NARMA model.

n	p	Measure	Percentage of training		
			50	65	80
100	10	$E(i)$	−2.334	−0.1958	−0.2934
		$std(i)$	1.5836	1.7273	1.7324
	15	$E(i)$	−0.7113	0.3311	−0.3016
		$std(i)$	1.8923	2.2368	1.5623
	25	$E(i)$	−0.3702	−0.316	*
		$std(i)$	1.5233	1.5139	*
200	10	$E(i)$	−0.6903	−0.339	−0.2041
		$std(i)$	1.7291	1.5689	1.6601
	15	$E(i)$	−0.3939	−0.3065	−0.3379
		$std(i)$	1.6468	1.6451	1.7154
	25	$E(i)$	−0.3945	−0.1167	−0.129
		$std(i)$	1.5177	1.5716	1.5575
	50	$E(i)$	−0.5299	−0.2362	*
		$std(i)$	1.756	1.4851	*
360	10	$E(i)$	−0.3284	−0.299	−0.346
		$std(i)$	1.6458	1.7647	1.5909
	15	$E(i)$	−0.3678	−0.371	−0.2601
		$std(i)$	1.559	1.5777	1.5391
	25	$E(i)$	−0.1201	−0.123	−0.2232
		$std(i)$	1.5785	1.5136	1.6092
	50	$E(i)$	−0.1744	−0.1713	−0.1388
		$std(i)$	1.2746	1.3208	−1.2823
	100	$E(i)$	−0.2222	**−0.05965**	*
		$std(i)$	1.06824	**1.01172**	*

percent was used to train the network. It is noted that although the NARMA network is not also capable of capturing all data behavior with performance of moving averages (see Fig. 7 graph (b)); it is found that using a lower number of parameters to be estimated has a better performance than the ARNN network.

5.2. Model 2

Table 4 contains the values of the normalized means square error (NMSE) found by Burges and Refenes [15] for NARMA models with 1, 2 and 3 lags (first three rows), and those obtained in this work by using the ARNN network and high order NARMA network with 1, 2 and 3 lags. The information for ARNN and NARMA models considered in this table is presented in Table 5, which contains information for each model of the performance measures suggested by Zemouri et al. [30]. The actual values of test versus the best forecasts of the networks ARNN AND NARMA are shown in Fig. 8.

Table 4
Comparison of results for simulated data model (11).

Model	Data of training	Data of validation	Data of proof
NARMA (1) [15]	0.813	0.846	NA
NARMA (2) [15]	0.692	0.755	NA
NARMA (3) [15]	0.689	0.789	NA
ARNN (10)	0.714	0.858	0.0858
ARNN (25)	0.636	0.864	**0.0198**
ARNN (50)	0.623	0.767	0.139
NARMA (1)	0.743	0.783	0.909
NARMA (2)	0.773	0.714	0.876
NARMA (3)	0.757	0.787	**0.855**

Table 5
Performance measures of the NARMA and ARNN models.

Model	$M1$	$M2$	$M3$	$M4$	$E(i)$	$std(i)$
ARNN (10)	0.115	1.999	0.134	0.445	−0.0394	1.0708
ARNN (25)	0.0904	1.852	0.15	0.478	**0.00544**	**1.101**
ARNN (50)	0.129	1.607	0.0565	0.558	−0.0417	0.153
NARMA (1)	−0.17	2.004	0.0276	0.537	−0.0841	1.89
NARMA (2)	−0.218	1.912	0.202	0.527	−0.0672	1.865
NARMA (3)	0.254	1.211	0.248	0.584	**−0.0249**	**1.852**

In Table 4 shows that the NARMA networks adjusted in this work, for each lag, have lower NMSE in the validation set than their corresponding found by Burges and Refenes [15]; for the case of ARNN networks, that none of them produced (under the validation set) a lower NMSE than best value found by the authors.

In the second experiment, it evidenced again that the problem of over parameterization of the ARNN networks have leading to inconsistency observed between NMSE values found for the three data sets (see Table 4). Following the proposed approach by Zemouri et al. [30], the best models are: ARNN (25) and NARMA (3). Note that there is a consistency to select the best model using NMSE or $E(i)$ measure (obtained for test data).

However, there is evidence that these models do not have a good predictive capability, given that in Fig. 8 clouds of points are far from the 45° line.

6. Discussion

In this section we answer the raised research questions.

1. *Can a nonlinear high order AR model, represented by ARNN network, be well approximated to nonlinear reduced order MA model*?

In examining whether the ARNN network with a high order for the lag p, is capable of approximating a NLMA correctly it found that while increasing the number of lags p, the MSE training tends to decrease (as shown in Fig. 6) and the measures $E(i)$ and $std(i)$ show better results, this fact is not reflected in the forecasting capacity of the model (Figure (a) of Fig. 7).

It is noteworthy that the forecast ability does not depend only on the value of the lag value, but the sample size and the data percentage used to train the network. The best results are obtained for ARNN networks with larger values of lags accompanied by large sample sizes, of which a large percentage is used for training. However, keep in mind that this leads to not adjust parsimonious or short term models and over parameterization problems.

If in addition to this, it is considered that NLMA model is not globally invertible, then the answer to the question is nonlinear autoregressive model (in this case approximated by an ARNN network) of a high order is not capable of representing a nonlinear moving averages model (NLMA) of low order.

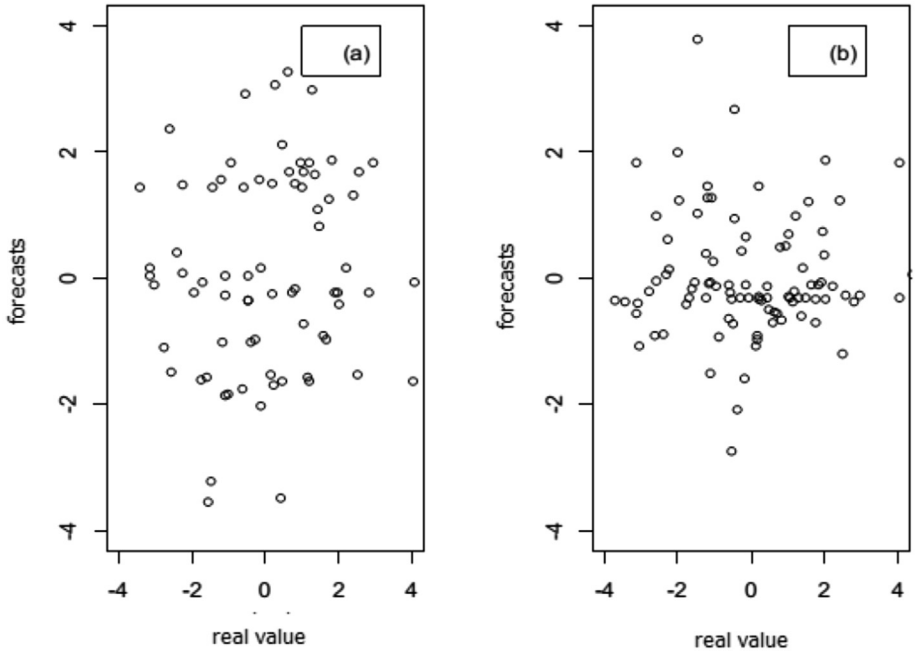

Fig. 8. Comparison between the test data and their forecasts found with the network (a) ARNN (25) and (b) NARMA (3).

2. *When in a NARMA that assumes there is no autoregressive process, can be predicted adequately a nonlinear time series containing inherent moving averages components*?

Figs. 7 and 8 and in Tables 2 and 5, it is observed that although the selected NARMA model has better performance (in terms of performance measures proposed by Zemouri et al. [30] approach to get straight 45°) than the other tested networks, the predicted values by this model are far from the actual values of the nonlinear series time with MA component. (See graphs (b) of Figs. 7 and 8). Considering this fact, the answer is that a recurrent network NARMA (0, q) cannot adequately predict nonlinear time series containing inherent moving averages components.

However, in testing it was noted that as is the case with mathematical expressions, practically NARMA network has a better approach to model NLMA (from the point of view of better forecasting capacity measures) than ARNN network. This indicates that this network can be a good candidate to nonlinear data model containing moving averages components, but requires to be studied in detail, and so a new research question arises: From the theoretical approach point of view, what are the considerations that the recurrent network NARMA (0, q) must have so it can predict properly nonlinear time series containing inherent moving average components?

7. Conclusion

It is shown that both the recurrent neural network NARMA model and autoregressive neural network ARNN model are unable to fully capture the behavior of a nonlinear time series containing inherent moving average (MA) component. This raises the need to develop an artificial neural network model or a hybrid model with Fuzzy Logic to adequately predict nonlinear time series with inherent MA component, which can have NARMA as a starting point.

References

[1] De Gooijer JG, Hyndman RJ. 25 years of time series forecasting. Int J Forecast 2006;22(3):443–73.
[2] Iacus SM. Statistical data analysis of financial time series and option pricing in R. Chicago: R/Finance, USA; 2011.
[3] Teräsvirta T. Forecasting economic variables with nonlinear models. SSE/EFI working paper in economics and finance. Stockholm: Department of Economic Statistics; 2005. p. 598.
[4] Engle Robert F. Risk and volatility: econometric models and financial practice. 44 West Fourth Street, New York, NY 10012-1126, USA.: New York University, Department of Finance (Salomon Centre); December 2003.
[5] van Dijk Dick, Medeiros Marcelo C, Teräsvirta Timo. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: a re-examination. Department of Economics PUC-Rio, Pontifical Catholic University of Rio de Janeiro Rua Marques de São Vicente 225-Rio de Janeiro 22453-900, RJ; 2004.
[6] Oscar Jorda Alvaro Escribano. Improved testing and specification of smooth transition regression models. Universidad Carlos III de Madrid; 1997.
[7] Robinson Peter M. Modelling memory of economic and financial time series. London School of Economics and Political Science; 2005.
[8] La Rocca Michele, Perna Cira. Model selection for neural network models: a statistical perspective. In: Emmert-Streib Frank, Matthias Dehmer Stefan Pickl, editors. Computational network theory: theoretical foundations and applications. 1st ed. Wiley-VCH Verlag GmbH & Co. KGaA; 2015.
[9] Pretorius Philip, Sibanda Wilbert. Artificial neural networks – a review of applications of neural networks in the modeling of HIV epidemic. Int J Comput Appl April 2012;44(16).
[10] Qi M, Zhang GP. An investigation of model selection criteria for neural network time series forecasting. Eur J Oper Res 2001;132(1):666–80.
[11] Lin CF, Granger CWJ, Teräsvirta T. Power of the neural network linearity test. J Time Ser Anal 1993;14(2):209–23.
[12] Bijari Mehdi, Hejazi Seyed Reza, Khashei Mehdi. Combining seasonal ARIMA models with computational intelligence techniques for time series forecasting. Soft Comput June 2012;16(6):1091–105. Springer.

[13] Qi Min, Zhang Peter G. Neural network forecasting for seasonal and trend time series. Eur J Oper Res February 2005;160(2):501–14.
[14] Benhra j, El Hassani H, Benkachcha S. Seasonal time series forecasting models based on artificial neural network. Int J Comput Appl April 2015; 116(20).
[15] Burges AN, Refenes A-PN. Modelling non-linear moving average processes using neural networks with error feedback: an application to implied volatility forecasting. Signal Process 1999;74(1):89–99.
[16] Connor JT, Martin RD. Recurrent neural networks and robust time series prediction. IEEE Trans Neural Netw 1994;5(2):240–53.
[17] Robinson PM. The estimation of a nonlinear moving average model. Stoch Process Their Appl 1977;5(1):81–90.
[18] Brännäs Kurt, Ohlsson Henry. Asymmetric time series and temporal aggregation. Rev Econ Stat May 1999;81(2):341–4.
[19] Robinson PM, Zaffaroni P. Modelling nonlinearity and long memory in time series. Fields Inst Commun 1997;11:161–70.
[20] Engle RF, Smith A. Stochastic permanent breaks. Rev Econ Stat 1999; 81(4):553–74.
[21] Kamil Scotto, Manuel González, de Zea Bermudez Turkman. Nonlinear time series models. Extreme events and integer value problems. 2014. p. 23–90.
[22] Chen D, Wang H. The stationarity and invertibility of a class of nonlinear ARMA models. Sci China Math March 2011;54(3):469–78.
[23] Haykin Simon O. Neural networks and learning machines. 3rd ed. 2006.
[24] Gencay R, Liu T. Nonlinear modelling and prediction with feedforward and recurrent networks. Phys D 1997;108(1–2):119–34.
[25] Stinchcombe M, White H, Hornik K. Multilayer feedforward networks are universal approximators. Neural Netw 1989;2(2):359–66.
[26] Stinchcombe M, White H, Homik K. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Netw 1990;3(5):551–60.
[27] Homik K. Approximation capabilities of multilayer feedforward networks. Neural Netw 1991;4(2):251.
[28] Rosa Joao Luis G. Artificial neural networks – models and applications. 2016.
[29] Patuwo BE, Hu MY, Zhang GP. A simulation study of artificial neural networks for nonlinear time series forecasting. Comput Oper Res 2001; 28(4):381–96.
[30] Gouriveau R, Zerhouni N, Zemouri R. Defining and applying prediction performance metrics on a recurrent NARX time series model. Neurocomputing 2010;73(13–15):2506–21.